arXiv: arXiv:1305.4152

Sparse Approximations in Spatio-Temporal Point Process Models

B. Cseke, A. Zammit-Mangion, G. Sanguinetti and T. Heskes

Address of B. Cseke and G. Sanguinetti
School of Informatics
10 Crichton Street
Edinburgh, EH8 9AB
United Kingdom
e-mail: bcseke@inf.ed.ac.uk; gsanguin@inf.ed.ac.uk

Address of A. Zammit-Mangion
School of Geographical Sciences
University Road
Clifton, Bristol BS8 1SS
United Kingdom
e-mail: a.zammitmangion@bristol.ac.uk

Address of T. Heskes
Faculty of Science
Heyendaalseweg 135
6525 AJ Nijmegen
The Netherlands
e-mail: t.heskes@science.ru.nl

Abstract: Analysis of spatio-temporal point patterns plays an important role in several disciplines, yet inference in these systems remains computationally challenging due to the high resolution modelling generally required by large data sets and the analytically intractable likelihood function. Here, we exploit the sparsity structure of a fully-discretised log-Gaussian Cox process model by using expectation constrained approximate inference. The resulting family of expectation propagation algorithms scale well with the state dimension and the length of the temporal horizon with moderate loss in distributional accuracy. They hence provide a flexible and faster alternative to both the filtering-smoothing type algorithms and the approaches which implement the Laplace method or expectation propagation on (block) sparse latent Gaussian models. We demonstrate the use of the proposed method in the reconstruction of conflict intensity levels in Afghanistan from a WikiLeaks data set.

Keywords and phrases: variational approximate inference, latent Gaussian models, sparse approximations, log-Gaussian Cox process.

1. Introduction

Spatio-temporal point-process modelling finds application in several fields such as epidemiology, [5], ecology [10] and criminology [20]. However, despite their

*Footnote to the title with the "thankstext" command.

importance and prevalence, inference in these systems remains computationally challenging. Markov chain Monte Carlo (MCMC) is frequently employed, however sampling is expensive and the problems under investigation are generally only of moderate size and complexity. On the other hand deterministic approximate inference methods for inference are rapidly gaining popularity in this field. These approaches can be classified as dynamic or static: The former class consists of filtering-smoothing type approaches such as the variational approach in [28] or the expectation propagation (EP) algorithm exploiting low rank approximations in [8]. The static approaches cast the discretised model as a latent (sparse) Gaussian block model and apply the Laplace method [21] or a corresponding EP algorithm [e.g. 2]. However, the computational scaling of these approaches is fixed and they may become untenable for large state dimensionality and long time series.

In this paper we derive an EP algorithm that reduces the computational complexity not by the above mentioned low rank approaches, but by exploiting the sparsity (in canonical/precision parameters) of the underlying latent Gaussian model. The approach we take exploits this sparsity in a similar way as in [21] and [2], however, the crucial step is the distribution of the computation. The expensive (sparse) linear algebraic operations are performed not on the (concatenated) block model, but instead on the two-time slice marginals characteristic to the dynamic approaches. The complexity of these local computations depends on the graph structure of the expectation constraints imposed on the approximate marginals, namely the sparsity structure of the Gaussian messages that these constraints result in. We introduce a class of constraints that result in messages having the following precision structures: (i) diagonal (factored messages), (ii) spanning tree (iii) chordal and finally (iv) fully connected (full messages). The latter corresponds to the standard filtering-smoothing type EP algorithm. An algorithmically similar approach to (i) for discrete Bayesian networks is presented in [15]. Comparisons on data generated from the model show that these algorithms scale well and depending on the complexity of the messages we can do approximate inference on hundreds or a few thousands of state variables and hundreds of time-steps with reasonable time and memory requirements.

This paper is structured as follows. In Section 2 we introduce the log-Gaussian Cox process and present the discretisation and approximation steps that simplify this model to a dynamic latent Gaussian model with non-Gaussian observations. In Section 3 we derive a class of dynamic EP algorithms that exploit sparse interaction structures. In Section 4 we discuss the performance of these algorithms and apply them to the WikiLeaks Afghan War Diary, a data set containing tens of thousands of events. The underlying system, known to experience micro-dynamic effects, can be modelled at a high resolution using one of the proposed approaches. Section 5 concludes the paper.

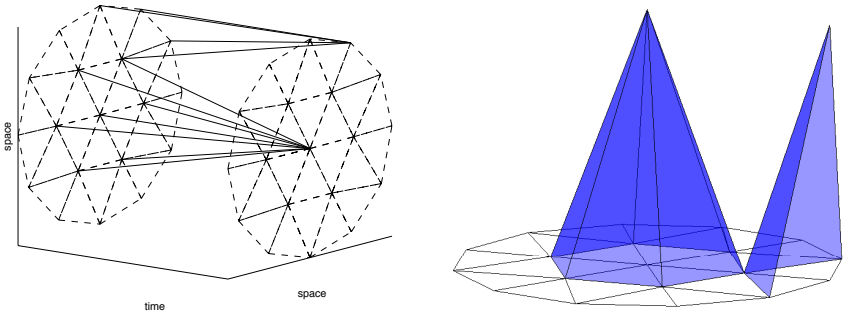

FIG 1. *An illustration of the spatio-temporal discretisation. The right panel illustrates two basis functions defined according to the triangular finite element spatial discretisation. The bases are shown for two nodes/vertices one in the interior and one on the boundary of the domain. The left panel illustrates some of the temporal connectivity for some nodes resulting from the spatio-temporal discretisation described in Section 2. Similarly, the temporal connectivities are shown only for an interior and a boundary node/vertex.*

2. Model and likelihood approximation

In this work we address point-processes with the following, underlying, spatio-temporal autoregressive system

$$\boldsymbol{x}_{t+1} = \boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{e}_t, \tag{2.1}$$

where t is a discrete temporal index, each $\boldsymbol{x}_t \in \mathbb{R}^n$, $\boldsymbol{e}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}^{-1})$ with both $\boldsymbol{A}$ and $\boldsymbol{Q}$ sparse. Equation (2.1) can be obtained from spatio-temporal models commonly employed in practice, such as the integro-difference equation (IDE) [26], and the stochastic partial differential equation (SPDE) [27]. Sparsity in $\boldsymbol{A}$ and $\boldsymbol{Q}$ ensues either by gridding the domain or by employing a Galerkin reduction on an infinite-dimensional system in $z_t(\boldsymbol{s}), \boldsymbol{s} \in \mathcal{O} \subset \mathbb{R}^2$, using basis functions of compact support. In both cases we can denote the approximate field as $z_t(\boldsymbol{s}) \approx \boldsymbol{\phi}(\boldsymbol{s})^T \boldsymbol{x}_t$ where $\{\boldsymbol{\phi}_i(\boldsymbol{s})\}_{i=1}^n$ is the basis; when using a grid the approximate field is discontinuous.

As is typical in spatio-temporal point-process applications, we model the intensity function of observed events as $\lambda_t(\boldsymbol{s}) = \exp(z_t(\boldsymbol{s}))$; in practice additional covariates may be included, however we omit these in order to facilitate the exposition. Let each observation window be of length Δ_t and $\mathcal{Y}_t = \{\boldsymbol{s}_i\}_{i \in I_t}$ where I_t is the set of indices corresponding to events in $(t-1, t]$, then the likelihood of each spatial point process is given by

$$\begin{aligned} p(\mathcal{Y}_t|\boldsymbol{x}_t) &\propto\!\!\propto \exp\left(-\Delta_t \int_{\mathcal{O}} e^{\boldsymbol{\phi}^T(\boldsymbol{s})\boldsymbol{x}_t} \mathrm{d}\boldsymbol{s}\right) \prod_{j \in I_t} e^{\boldsymbol{\phi}^T(\boldsymbol{s}_j)\boldsymbol{x}_t} \\ &= L_1(\boldsymbol{x}_t) L_2(\boldsymbol{x}_t; \mathcal{Y}_t) \end{aligned} \tag{2.2}$$

This likelihood can be split into two components; the first ($L_1(\boldsymbol{x}_t)$) is directly related to the *void probability* of the process. We adopt the approach in [23] and approximate the integral as:

$$\begin{aligned}\log L_1(\boldsymbol{x}_t) &\approx -\Delta_t \sum_{i=1}^{p} \widetilde{\eta}_i \exp(\boldsymbol{\phi}^T(\bar{\boldsymbol{s}}_i)\boldsymbol{x}_t) \\ &= -\boldsymbol{\eta}^T \exp(\boldsymbol{C}_1 \boldsymbol{x}_t),\end{aligned} \tag{2.3}$$

where $\boldsymbol{\eta} = \Delta_t \widetilde{\boldsymbol{\eta}}$ are the scaled integration weights and $\boldsymbol{C}_1 = [\boldsymbol{\phi}(\bar{\boldsymbol{s}}_1) \dots \boldsymbol{\phi}(\bar{\boldsymbol{s}}_p)]^T$ contains the values of the basis at the chosen p integration points $\{\bar{\boldsymbol{s}}_i\}_{i=1}^p$. The second component of the likelihood, $L_2(\boldsymbol{x}_t; \mathcal{Y}_t)$, adds contributions from the observed events and can be represented as follows

$$\log L_2(\boldsymbol{x}_t; \mathcal{Y}_t) = \sum_{j \in I_t} \boldsymbol{\phi}^T(\boldsymbol{s}_j)\boldsymbol{x}_t = \mathbf{1}^T \boldsymbol{C}_2(\mathcal{Y}_t)\boldsymbol{x}_t, \tag{2.4}$$

where $\boldsymbol{C}_2(\mathcal{Y}_t)^T = [\boldsymbol{\phi}(\boldsymbol{s}_i)]_{i \in I_t}$. The log-likelihood can hence be written, up to a proportionality constant, as

$$\log p(\mathcal{Y}_t | \boldsymbol{x}_t) \approx -\boldsymbol{\eta}^T \exp(\boldsymbol{C}_1 \boldsymbol{x}_t) + \mathbf{1}^T \boldsymbol{C}_2(\mathcal{Y}_t)\boldsymbol{x}_t. \tag{2.5}$$

Both compact basis functions and gridded domains induce sparsity into the observation matrices $\boldsymbol{C}_1$ and $\boldsymbol{C}_2$. In particular, if one chooses the integration points to be the vertices of a triangulation or the centres of gridded cells, then $\boldsymbol{C}_1$ simplifies to the identity matrix $\boldsymbol{I}_{n \times n}$ where $n = p$. The integration weights $\widetilde{\boldsymbol{\eta}}$ then correspond to the volumes of the basis with unit weight. In addition, $\boldsymbol{C}_2(\mathcal{Y}_t)$ is again sparse with at most one non-zero element in each row for the gridded case and three non-zero elements per row for the triangulated case.

The spatial discretisation results in a latent Gaussian model. Since $\boldsymbol{C}_1$ is the identity and (2.4) is linear, the non-Gaussian terms will depend on x_{t+1}^j only and from (2.3) we define the proxy $\psi_{t+1,j}(x_{t+1}^j) = \exp\{-\eta_j \exp(x_{t+1}^j)\}$. Letting $\boldsymbol{X} = \{\boldsymbol{x}_t\}_{t=1}^T$ and $\mathcal{Y} = \{\mathcal{Y}_t\}_{t=1}^T$, the latent Gaussian model is given by

$$p(\boldsymbol{X}|\mathcal{Y}) \propto p_1(\boldsymbol{x}_1) \prod_t N(\boldsymbol{x}_{t+1} | \boldsymbol{A}\boldsymbol{x}_t, \boldsymbol{Q}^{-1}) \times \exp(\boldsymbol{x}_{t+1} \cdot \boldsymbol{h}_{t+1}^y) \prod_j \psi_{t+1,j}(x_{t+1}^j),$$

where $\boldsymbol{h}_{t+1}^y = \mathbf{1}^T \boldsymbol{C}_2(\mathcal{Y}_{t+1})$.

3. Inference

We define the factors[1] $\Psi_{t,t+1}(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}) = N(\boldsymbol{x}_{t+1} | \boldsymbol{A}\boldsymbol{x}_t, \boldsymbol{Q}^{-1}) \exp(\boldsymbol{x}_{t+1} \cdot \boldsymbol{h}_{t+1}^y)$–considering $t = 1$ as a special case including both the starting conditions and observations at $t = 1$–and write

$$p(\boldsymbol{X}|\mathcal{Y}) \propto \prod_t \Psi_{t,t+1}(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}) \prod_j \psi_{t+1,j}(x_{t+1}^j) \tag{3.1}$$

[1] Because of the abundance of indices, we use "$\cdot$" as a proxy for the inner product.

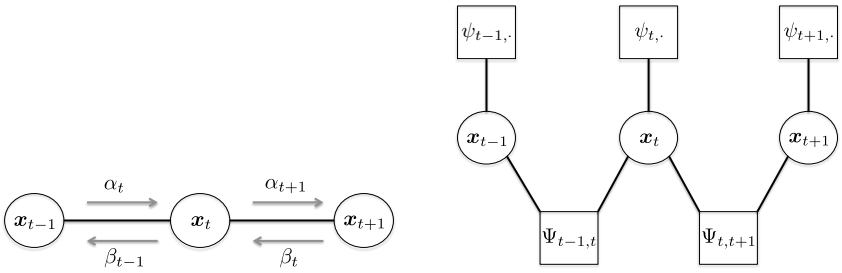

FIG 2. *An illustration of the chain graphical model (left panel) and the factor graph (right panel) corresponding to the model in* (3.1).

Note that p can be viewed as a (block) sparse latent Gaussian model and there are various approaches that approximate p and by a (block) Gaussian and apply corrections to it's marginals: (i) the Laplace method and marginal corrections [21] (ii) expectation propagation and marginal corrections [17, 2] and (iii) the standard variational approximation (no corrections), see [16]. It is generally known that both EP and the variational approach outperform the Laplace method and EP has a computational complexity that scales with that of the Laplace method [11, 2].

However, since we have a dynamical model, we will try to exploit its structure instead of viewing it as a generic sparse latent Gaussian model. Since we are dealing with a chain structured time series model, inference could be done by using the standard forward-backward message passing equations

$$\alpha_{t+1}(\boldsymbol{x}_{t+1}) \propto \int d\boldsymbol{x}_t \alpha_t(\boldsymbol{x}_t) \Psi_{t,t+1}(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}) \prod_j \psi_{t+1,j}(x^j_{t+1}) \tag{3.2}$$

and

$$\beta_t(\boldsymbol{x}_t) \propto \int d\boldsymbol{x}_{t+1} \beta_{t+1}(\boldsymbol{x}_{t+1}) \Psi_{t,t+1}(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}) \prod_j \psi_{t+1,j}(x^j_{t+1}) \tag{3.3}$$

and forming the one and two time-slice marginals

$$q(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}) \propto \beta_{t+1}(\boldsymbol{x}_{t+1}) \alpha_t(\boldsymbol{x}_t) \Psi_{t,t+1}(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}) \prod_j \psi_{t+1,j}(x^j_{t+1}). \tag{3.4}$$

Unfortunately, these computations are not tractable because: (i) both the forward and backward message passing involve multivariate integrals that cannot be computed analytically, (ii) in the cases we address the dimension of the state space scales with 10^3 and thus even the Gaussian case becomes computationally unfeasible. To address this problem we propose approximations that result in (i) propagating Gaussian messages α_t and β_t (ii) we restrict the precision structure of these Gaussian messages to make use of the sparsity of $\boldsymbol{A}$ and $\boldsymbol{Q}$ in the

parametric expression of q in (3.5) and thus all the matrix algebra we perform will be carried out on sparse matrices.

In the following we present the basic idea behind the approximation we propose. Since the messages are not exact, one forward-backward cycle is not sufficient to obtain the marginals and thus we have to turn (3.2) and (3.3) into an iterative procedure where several message updates have to be performed. In order to define the updates, we make use of (3.5). By multiplying (3.2) by $\beta_{t+1}(\boldsymbol{x}_{t+1})$ we obtain

$$\beta_{t+1}(\boldsymbol{x}_{t+1})\alpha_{t+1}(\boldsymbol{x}_{t+1}) \propto \int d\boldsymbol{x}_t \, q(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}). \tag{3.5}$$

We will approximate $q(\boldsymbol{x}_{t+1})$ from the equation above with a Gaussian having a restricted precision structure. Let us denote it by, say, $\tilde{q}(\boldsymbol{x}_{t+1})$ and define an iterative update procedure

$$\alpha^{new}_{t+1}(\boldsymbol{x}_{t+1}) \propto \tilde{q}(\boldsymbol{x}_{t+1})/\beta_{t+1}(\boldsymbol{x}_{t+1}) \quad \text{and} \quad \beta^{new}_{t+1}(\boldsymbol{x}_{t+1}) \propto \tilde{q}(\boldsymbol{x}_{t+1})/\alpha_{t+1}(\boldsymbol{x}_{t+1}),$$

where $\tilde{q}(\boldsymbol{x}_{t+1})$ is computed by approximating the marginal of $q(\boldsymbol{x}_t, \boldsymbol{x}_{t+1})$ which in turn is recomputed according to (3.5). We iterate these updates until the changes in α_t and β_t are within a predefined accuracy level.

3.1. Variational inference with expectation constraints

In order to define the approximation $\tilde{q}(\boldsymbol{x}_{t+1})$ and to formulate our approach in a sound methodological framework, we resort to the so called variational free energy approach in [9]. The main idea of this approach is that instead of approximating p with a Gaussian q, it only aims to approximate its marginals. It defines a family of marginals that, as consistency criteria, are assumed to satisfy a set of expectation constraints and then optimises them by finding a stationary point of a Kullback-Leibler divergence based variational objective. In the following we show how and under what limitations this approach can be applied to fit our requirements. Following standard variational approaches [e.g. 9], we use the KL-divergence $\mathrm{D}[q||p]$ and its approximations as means to approximate the marginals of p. We start from the standard variational approach, where due to the factorisation in p, we have

$$\mathrm{D}[q||p] = -\sum_t \mathrm{E}_q[\log \Psi_{t,t+1}] - \sum_{t,j} \mathrm{E}_q[\log \psi_{t+1,j}] - H(q) + \log Z_p, \tag{3.6}$$

where $H(q)$ is the entropy of q and $\log Z_p$ is the (unknown) normalisation constant of p. Note that minimising (3.6) w.r.t. a q restricted to be Gaussian leads to the standard (block) variational approximation (iii) mentioned above.

To exploit the decomposition of $\mathrm{D}[q||p]$ we define a family of approximate marginals $\mathcal{Q} = \left\{\{q^g_{t,t+1}\}_t, \{q^{gs}_t\}_t, \{q^l_{t+1,j}\}_{t,j}, \{q^{ls}_{t+1,j}\}_{t,j}\right\}$ which can be viewed as

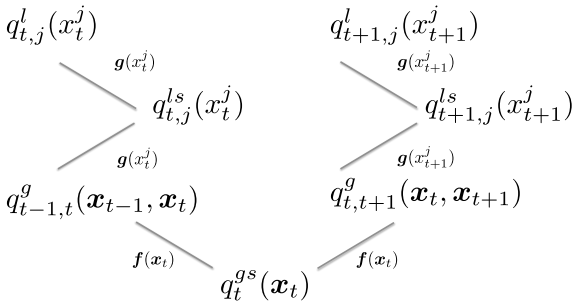

FIG 3. *An illustration of the approximate marginals in $\mathcal{Q}$ and the expectation constraints defined over then. The figure also illustrates the relations embedded in the tree representation in (3.7).*

corresponding to the un-normalized tree-structured density

$$q(\boldsymbol{X}) \propto \frac{\prod_{t=1}^{T-1} q^g_{t,t+1}(\boldsymbol{x}_t,\boldsymbol{x}_{t+1}) \prod_{t=1}^{T}\prod_j q^l_{t+1,j}(x^j_{t+1})}{\prod_{t=2}^{T-1} q^{gs}_t(\boldsymbol{x}_t) \prod_{t=1}^{T}\prod_j q^{ls}_{t+1,j}(x^j_{t+1})}. \tag{3.7}$$

An illustration of the structure of (3.7) is shown on Figure 3. All densities in $\mathcal{Q}$ are approximations of the corresponding marginals and, as we will see later, the optimal marginals lead to a q for which $q(\boldsymbol{X}) \propto p(\boldsymbol{X}|\mathcal{Y})$ holds. In particular, $q^g_{t,t+1}$ is assigned to the factor $\Psi_{t,t+1}$ and $q^l_{t+1,j}$ is assigned to $\psi_{t+1,j}$, whereas q^{gs}_t and $q^{ls}_{t+1,j}$ correspond to the separator densities used in graphical models [e.g. 12]. Let $\boldsymbol{f}(\boldsymbol{z})$ be the sufficient statistic of a (sparse) Gaussian with sparsity structure $G(\boldsymbol{f})$—to be defined later—and let $\boldsymbol{g}(z) = (z, -z^2/2)$ to denote the sufficient statistic of the univariate Gaussian. We use first and second order statistics as consistency criteria in the approximate marginals, and thus, we define the (temporal) expectation constraints

$$\mathrm{E}_{q^g_{t,t+1}}[\boldsymbol{f}(\boldsymbol{x}_{t+1})] = \mathrm{E}_{q^{gs}_{t+1}}[\boldsymbol{f}(\boldsymbol{x}_{t+1})] \quad \text{and} \quad \mathrm{E}_{q^g_{t+1,t+2}}[\boldsymbol{f}(\boldsymbol{x}_{t+1})] = \mathrm{E}_{q^{gs}_{t+1}}[\boldsymbol{f}(\boldsymbol{x}_{t+1})] \tag{3.8}$$

as well as the (spatial) constraints

$$\mathrm{E}_{q^l_{t+1}}\left[\boldsymbol{g}(x^j_{t+1})\right] = \mathrm{E}_{q^{ls}_{t+1}}\left[\boldsymbol{g}(x^j_{t+1})\right] \quad \text{and} \quad \mathrm{E}_{q^g_{t+1}}\left[\boldsymbol{g}(x^j_{t+1})\right] = \mathrm{E}_{q^{ls}_{t+1}}\left[\boldsymbol{g}(x^j_{t+1})\right]. \tag{3.9}$$

When $\boldsymbol{f}$ corresponds to a fully connected Gaussian, the corresponding temporal expectation constraints are marginal matching constraints. We derive our approach for a general $\boldsymbol{f}$ and we discuss possible choices in Sections 3.3 and 3.4.

By making use of the tree structure of q, we define the entropy approximation corresponding to $\mathcal{Q}$ by

$$-\tilde{H}(\mathcal{Q}) = \sum_t \mathrm{E}_{q^g_{t,t+1}}\left[\log q^g_{t,t+1}\right] - \sum_t \mathrm{E}_{q^{gs}_t}[\log q^{gs}_t] + \sum_{t,j}[\mathrm{E}_{q^l_{t+1,j}}\left[\log q^l_{t+1,j}\right] - \mathrm{E}_{q^{ls}_{t,j+1}}\left[\log q^{ls}_{t+1,j}\right]].$$

Note, that due to the tree structure, when the members of $\mathcal{Q}$ are true marginals, $\tilde{H}(\mathcal{Q})$ is equal to the true/exact entropy. We approximate $\mathrm{E}_q[\log p]$ by using the corresponding members of $\mathcal{Q}$ and arrive to an approximation (without the constant $\log Z_p$) of the variational objective (free energy) in (3.6) which reads as

$$F(\mathcal{Q}) = -\sum_t \mathrm{E}_{q^g_{t,t+1}}[\log \Psi_{t,t+1}] - \sum_{t,j} \mathrm{E}_{q^l_{t+1,j}}[\log \psi_{t+1,j}] - \tilde{H}(\mathcal{Q}). \tag{3.10}$$

To deal with the expectation constraints, we introduce the Lagrange multipliers $\boldsymbol{\alpha}_{t+1}$ and $\boldsymbol{\beta}_{t+1}$ for the (temporal) constraints w.r.t. $q^g_{t+1,t+2}$ and q^{gs}_{t+1}, and $q^g_{t,t+1}$ and q^{gs}_{t+1}, respectively. The multipliers corresponding to the (spatial) constraints on $q^g_{t,t+1}$ and $q^{ls}_{t+1,j}$, and $q^l_{t+1,j}$ and $q^{ls}_{t+1,j}$ will be denoted by $\boldsymbol{\lambda}^g_{t+1,j}$ and $\boldsymbol{\lambda}^l_{t+1,j}$, respectively. The stationary conditions of the Lagrangian corresponding to $F(\mathcal{Q})$ in (3.10) and the expectation constraints (3.8)-(3.9) result in the densities

$$q^g_{t,t+1}(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}) \propto \Psi_{t,t+1}(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}) \exp\{\textstyle\sum_j \boldsymbol{\lambda}^g_{t+1,j} \cdot \boldsymbol{g}(x^j_{t+1})\} \times \exp\left\{\boldsymbol{\alpha}_t \cdot \boldsymbol{f}(\boldsymbol{x}_t) + \boldsymbol{\beta}_{t+1} \cdot \boldsymbol{f}(\boldsymbol{x}_{t+1})\right\}, \tag{3.11}$$

$$q^{gs}_{t+1}(\boldsymbol{x}_{t+1}) \propto \exp\left\{(\boldsymbol{\alpha}_{t+1} + \boldsymbol{\beta}_{t+1}) \cdot \boldsymbol{f}(\boldsymbol{x}_{t+1})\right\}, \tag{3.12}$$

$$q^l_{t+1,j}(x^j_{t+1}) \propto \psi_{t+1,j}(x^j_{t+1}) \exp\{\boldsymbol{\lambda}^l_{t+1,j} \cdot \boldsymbol{g}(x^j_{t+1})\}, \tag{3.13}$$

$$q^{ls}_{t+1,j}(x^j_{t+1}) \propto \exp\{(\boldsymbol{\lambda}^g_{t+1,j} + \boldsymbol{\lambda}^l_{t+1,j}) \cdot \boldsymbol{g}(x^j_{t+1})\}. \tag{3.14}$$

Since all q^{gs}_{t+1} and $q^{ls}_{t+1,j}$ are Gaussians, the stationary conditions corresponding to the expectation constraints (moment matching) in (3.8) and (3.9) can be rewritten in terms of natural or canonical parameters as

$$\text{Collapse}\left[q^g_{t,t+1}(\boldsymbol{x}_{t+1}); \boldsymbol{f}\right] = \boldsymbol{\alpha}_{t+1} + \boldsymbol{\beta}_{t+1}, \tag{3.15}$$

$$\text{Collapse}\left[q^g_{t+1,t+2}(\boldsymbol{x}_{t+1}); \boldsymbol{f}\right] = \boldsymbol{\alpha}_{t+1} + \boldsymbol{\beta}_{t+1}, \tag{3.16}$$

$$\text{Collapse}\left[q^g_{t,t+1}(x^j_{t+1}); \boldsymbol{g}\right] = \boldsymbol{\lambda}^g_{t+1,j} + \boldsymbol{\lambda}^l_{t+1,j}, \tag{3.17}$$

$$\text{Collapse}\left[q^l_{t+1,j}(x^j_{t+1}); \boldsymbol{g}\right] = \boldsymbol{\lambda}^g_{t+1,j} + \boldsymbol{\lambda}^l_{t+1,j}. \tag{3.18}$$

Here, Collapse$[q; \boldsymbol{f}]$ and Collapse$[q; \boldsymbol{g}]$ are the projections of q into the Gaussian families defined by $\boldsymbol{f}$ and $\boldsymbol{g}$ respectively. In other words, suppose $S(\hat{\boldsymbol{Q}})$ denores the sparsity structure of the matrix $\hat{\boldsymbol{Q}}$, then

$$\text{Collapse}[q; \boldsymbol{f}] \equiv \underset{(\hat{\boldsymbol{h}}, \hat{\boldsymbol{Q}}): S(\hat{\boldsymbol{Q}}) = G(\boldsymbol{f})}{\text{argmin}} \mathrm{D}\left[q(\boldsymbol{z}) || N(\boldsymbol{z}; \hat{\boldsymbol{Q}}^{-1}\hat{\boldsymbol{h}}, \hat{\boldsymbol{Q}}^{-1})\right]. \tag{3.19}$$

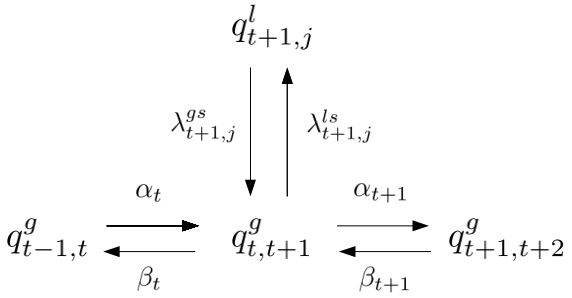

FIG 4. *Illustration of the message passing inference algorithm.*

By turning (3.15) and (3.16) into a pair of update formulae, we arrive at forward-backward messages similar to (3.2) and (3.3): $\boldsymbol{\alpha}_{t+1}$ is the canonical parameter of the forward message $\alpha_{t+1}(\boldsymbol{x}_{t+1})$ and $\boldsymbol{\beta}_{t+1}$ is the canonical parameter of the backward message $\beta_{t+1}(\boldsymbol{x}_t)$. In a similar fashion, the messages corresponding to (3.17) and (3.18) correspond to the messages of an expectation propagation algorithm in a latent Gaussian model: $\boldsymbol{\lambda}^g_{t+1,j}$ corresponds to the parameters of the term approximation while $\boldsymbol{\lambda}^l_{t+1,j}$ corresponds to the parameters of the so-called cavity distribution [18, 14]. As a result, the message passing algorithm can be written as

$$\boldsymbol{\alpha}^{new}_{t+1} = \text{Collapse}\left[q^g_{t,t+1}(\boldsymbol{x}_{t+1}); \boldsymbol{f}\right] - \boldsymbol{\beta}_{t+1}, \tag{3.20}$$

$$\boldsymbol{\beta}^{new}_{t} = \text{Collapse}\left[q^g_{t,t+1}(\boldsymbol{x}_{t}); \boldsymbol{f}\right] - \boldsymbol{\alpha}_{t}, \tag{3.21}$$

$$\left[\boldsymbol{\lambda}^l_{t+1,j}\right]^{new} = \text{Collapse}[q^g_{t,t+1}(x^j_{t+1}); \boldsymbol{g}] - \boldsymbol{\lambda}^g_{t+1,j}, \tag{3.22}$$

$$\left[\boldsymbol{\lambda}^g_{t+1,j}\right]^{new} = \text{Collapse}[q^l_{t+1,j}(x^j_{t+1}); \boldsymbol{g}] - \boldsymbol{\lambda}^l_{t+1,j}. \tag{3.23}$$

Appropriate damping of the form $x^{new} = (1-\epsilon)x^{old} + \epsilon x^{new}$ might need to be applied to help convergence [9]. We update $\boldsymbol{\lambda}^g_{t+1,j}$ and $\boldsymbol{\lambda}^l_{t+1,j}$ for all j simultaneously as in [2].

The message passing updates from above combine both temporal and spatial inference methods in a simple way. They are suited to distributed computations and, as we show in the next section, are more suited to exploit the sparsity of $\boldsymbol{A}$ and $\boldsymbol{Q}$ than the typical Kalman filtering and R.T.S. smoothing methods that do inference using prediction (marginalisation) and innovation (Bayesian update) steps, resulting in operations on full matrices.

In retrospect, we provided solutions to the problems raised in the first part of Section 3 in the following way: (i) We worked around the the numerical intractability in (3.2) and (3.3) by separating the non-Gaussian terms $\psi_{t+1,j}(x^j_{t+1})$ from $\Psi_{t,t+1}(\boldsymbol{x}_t, \boldsymbol{x}_{t+1})$ through the use of expectation constraints and thus the marginalization is performed on the tractable $q^g_{t,t+1}(\boldsymbol{x}_t, \boldsymbol{x}_{t+1})$. As mentioned earlier, the updates (3.22) and (3.23) implement the expectation propagation procedure in a latent Gaussian model with sparse precision structure and thus

$q^g_{t,t+1}(\boldsymbol{x}_t, \boldsymbol{x}_{t+1})$ is an accurate proxy for providing the required moments [2]. (ii) We showed that expectation constrains provide a natural way to introduce a message passing algorithm where the messages are Gaussian having canonical parameters with sparse precision structure $G(\boldsymbol{f})$; this relation provides a good intuition for choosing the sparsity structure. In the following section we provide a detailed discussion of when and how the above message passing can be done in a computational efficient way.

3.2. Messages and collapse operations

As can be seen from (3.20)-(3.23), the sufficient statistics (structures) of the messages are defined by the expectation constraints, that is, by the Gaussian families defined by $\boldsymbol{f}$ and $\boldsymbol{g}$. In the following we present the details of collapse steps in these equations. These steps consist of computing the required first and second order moments of the approximate marginals $q^g_{t,t+1}$ and $q^l_{t+1,j}$ and using them to project the approximate marginals—according to (3.19)—into the canonical Gaussian families defined by $\boldsymbol{f}$ and $\boldsymbol{g}$ respectively.

The Collapse$[q^l_{t+1,j}(x^j_{t+1}); \boldsymbol{g}]$ step in (3.23) can be performed as follows. Since $\psi_{t+1,j}$ depends only on x^j_{t+1}, we have to compute the first and second moments of z, where z is distributed according to an un-normalised distribution $\psi_{t+1,j}(z)\exp(hz - qz^2/2)$, where h and q are the canonical parameters in $\boldsymbol{\lambda}^l_{t+1,j}$. Since the computation of the moments cannot be carried out analytically (see (2.3)) one has to resort to numerical approximations. We propose two alternatives: (i) applying Gauss-Hermite numerical quadrature w.r.t. z or (ii) finding the Gaussian approximation of $\psi_{t+1,j}(z)\exp(hz - qz^2/2)$ by the univariate Laplace method and performing Gauss-Hermite numerical quadrature w.r.t. this approximation. Because of the univariate Newton method we use, method (ii) is slightly more computationally expensive than method (i), however, it is more accurate when the masses of $\psi_{t+1,j}(z)$ and $\exp(hz - qz^2/2)$ are far apart as it is often the case in the first cycle of updates. Due to the accuracy of these univariate methods, the numerical error in computing the moments and thus Collapse$[q^l_{t+1,j}(x^j_{t+1}); \boldsymbol{g}]$ is negligible.

The Collapse$[q^g_{t,t+1}(x^j_{t+1}); \boldsymbol{g}]$ step in (3.22) reduces to the computation of the marginal means and variances $q^g_{t,t+1}(x^j_{t+1})$. This requires a marginalisation which can be computationally expensive. The crucial ides that leads to significant computational savings is that, in order to preserve sparsity, we carry out the computations on the joint $q^g_{t,t+1}(\boldsymbol{x}_t, \boldsymbol{x}_{t+1})$ by using sparse partial matrix inversion via sparse Cholesky factorisation and solving the corresponding Takahashi equations [24]. Let $\boldsymbol{\alpha}_t = (\boldsymbol{h}_{\boldsymbol{\alpha}_t}, \boldsymbol{Q}_{\boldsymbol{\alpha}_t})$ and $\boldsymbol{\beta}_{t+1} = (\boldsymbol{h}_{\boldsymbol{\beta}_{t+1}}, \boldsymbol{Q}_{\boldsymbol{\beta}_{t+1}})$ denote the canonical representations of $\boldsymbol{\alpha}_t$ and $\boldsymbol{\beta}_{t+1}$, and let us concatenate $\boldsymbol{\lambda}^g_{t+1,j}$ into the representation $\boldsymbol{\lambda}^g_{t+1,\cdot} = (\boldsymbol{h}_{\boldsymbol{\lambda}^g_{t+1,\cdot}}, \boldsymbol{Q}_{\boldsymbol{\lambda}^g_{t+1,\cdot}})$ where, due to the univariate nature of $\boldsymbol{\lambda}^g_{t+1,j}$s the precision parameter $\boldsymbol{Q}_{\boldsymbol{\lambda}^g_{t+1,\cdot}}$ is diagonal. With this notation, the

precision matrix of $q^g_{t,t+1}$ can be written as

$$\boldsymbol{Q}^g_t = \begin{bmatrix} \boldsymbol{A}^T\boldsymbol{Q}\boldsymbol{A} + \boldsymbol{Q}_{\boldsymbol{\alpha}_t} & -\boldsymbol{A}^T\boldsymbol{Q} \\ -\boldsymbol{Q}\boldsymbol{A} & \boldsymbol{Q} + \boldsymbol{Q}_{\boldsymbol{\beta}_{t+1}} + \boldsymbol{Q}_{\boldsymbol{\lambda}^g_{t+1,\cdot}} \end{bmatrix}, \tag{3.24}$$

where $\boldsymbol{Q}_{\boldsymbol{\alpha}_t}$, $\boldsymbol{Q}_{\boldsymbol{\beta}_{t+1}}$ and $\boldsymbol{Q}_{\boldsymbol{\lambda}^g_{t+1,\cdot}}$ are the precision terms corresponding to the messages $\boldsymbol{\alpha}_t$, $\boldsymbol{\beta}_{t+1}$ and $\boldsymbol{\lambda}^g_{t+1,\cdot}$ and the rest of the parameters correspond to the Gaussian transition probability $\Psi_{t,t+1}$. The sparsity of $\boldsymbol{Q}^g_t$ is mainly determined by the sparsity of $\boldsymbol{A}^T\boldsymbol{Q}\boldsymbol{A}$ and the sparsity of the messages precision structure $\boldsymbol{Q}_{\boldsymbol{\alpha}_t}$ and $\boldsymbol{Q}_{\boldsymbol{\beta}_{t+1}}$, that is, by $G(\boldsymbol{f})$. To compute the required moments we (i) solve the system $[\boldsymbol{Q}^g_t]^{-1}[\boldsymbol{h}^g_t]$, where $\boldsymbol{h}^{g^T}_t = [\boldsymbol{h}^T_{\boldsymbol{\alpha}_t}, \boldsymbol{h}^T_{\boldsymbol{\beta}_{t+1}} + \boldsymbol{h}^{y^T}_{t+1} + \boldsymbol{h}^T_{\boldsymbol{\lambda}^g_{t+1,\cdot}}]$ and (ii) compute the diagonal of $[\boldsymbol{Q}^g_t]^{-1}$. We do this by a sparse Cholesky factorisation of a convenient reordering of $\boldsymbol{Q}^g_t$ followed by (i) solving the linear system and (ii) doing a partial inversion by solving the Takahashi equations. Let $\boldsymbol{L}\boldsymbol{L}^T = [\boldsymbol{Q}^g_t]_{\sigma,\sigma}$ where σ is the permutation corresponding to a reordering. Then the partial inversion using the Takahashi equations computes all entries of $[[\boldsymbol{Q}^g_t]_{\sigma,\sigma}]^{-1}$ for which $\boldsymbol{L}$ is non-zero. This implies that all entries of $[\boldsymbol{Q}^g_t]^{-1}$ where $\boldsymbol{Q}^g_t$ is non-zero are computed [4]—a property which will be further exploited in the Collapse$[\cdot;\boldsymbol{f}]$ step. The partial matrix inversion can be viewed as running a junction tree algorithm on the Gaussian graphical model defined by $\boldsymbol{Q}^g_t$, where the junction tree is constructed by the sparse Cholesky factorisation [4].

The Collapse$[q^g_{t,t+1}(\boldsymbol{x}_{t+1});\boldsymbol{f}]$ and Collapse$[q^g_{t,t+1}(\boldsymbol{x}_t);\boldsymbol{f}]$ steps in (3.20) and (3.21) compute the temporal messages and can be performed as follows. Let $q(\boldsymbol{x}) = N(\boldsymbol{x}|\boldsymbol{m},\boldsymbol{V})$, then according to (3.19), Collapse$[q(\boldsymbol{x});\boldsymbol{f}]$ simplifies to solving

$$\begin{aligned} \underset{\hat{\boldsymbol{Q}}}{\text{minimise}} \quad & \operatorname{tr}\left(\boldsymbol{V}\hat{\boldsymbol{Q}}\right) - \log\det\hat{\boldsymbol{Q}} \\ \text{s.t.} \quad & \hat{Q}_{i,j} = 0, \text{ for all } (i,j) \notin G(\boldsymbol{f}), \end{aligned}$$

and computing $\hat{\boldsymbol{h}} = \hat{\boldsymbol{Q}}^{-1}\boldsymbol{m}$. The optimisation can be solved by gradient based methods or the Newton method, however, when the graph $G(\boldsymbol{f})$ is chordal, the optimality conditions lead to equations that can be solved exactly (instead of expensive optimisation) by using the values $\boldsymbol{V}_{ij}$ with $(i,j) \in G(\boldsymbol{f})$ [3]. Since the covariance values corresponding to the non-zeros in $\boldsymbol{Q}_{\boldsymbol{\alpha}_t}$ and $\boldsymbol{Q}_{\boldsymbol{\beta}_{t+1}}$ are all already computed by the partial matrix inversion of $\boldsymbol{Q}^g_t$ no further covariance computations are needed. This leads to significant computational advantages. For chordal $G(\boldsymbol{f})$s the equations for the optimality conditions can be solved as follwos [3]. Let $C_1,\ldots,C_K$ be the cliques of $G(\boldsymbol{f})$. Assume further that the cliques of the graph's junction tree are ordered such that if C_i is an ancestor of C_j then $i \leq j$. Let $S_j = C_j \cap (C_1 \cup C_2 \cup \ldots \cup C_{j-1})$ and $R_j = C_j \setminus (C_1 \cup C_2 \cup \ldots \cup C_{j-1})$. Then $\hat{\boldsymbol{Q}} = (\boldsymbol{I}+\boldsymbol{U})\boldsymbol{D}(\boldsymbol{I}+\boldsymbol{U})^T$, where $\boldsymbol{U}$ and $\boldsymbol{D}$ can be computed iteratively from

$$\boldsymbol{U}_{R_k,S_k} = -\boldsymbol{V}^{-1}_{S_k,S_k}\boldsymbol{V}_{S_k,R_k}, \tag{3.25}$$

$$\boldsymbol{D}_{R_k,R_k} = [\boldsymbol{V}_{R_k,R_k} - \boldsymbol{V}_{R_k,S_k}\boldsymbol{V}^{-1}_{S_k,S_k}\boldsymbol{V}_{S_k,R_k}]^{-1}. \tag{3.26}$$

The computational complexity scales with $\sum_k \max\{|S_k|^3, |R_k|^3, |S_k|^2|R_k|\}$. The size of the cliques depends on the structure of $G(\boldsymbol{f})$, see Section 3.4 for further details.

The above algorithm can be given the following intuitive interpretation. Suppose again that we want to minimise the KL-divergence $\mathrm{D}[q(\boldsymbol{x})||p(\boldsymbol{x})]$ w.r.t. p assuming that p is a tree structured distribution according to the clique structure $C_1, \ldots, C_K$. Then simple calculus yields that $p(\boldsymbol{x}_{C_k}) = q(\boldsymbol{x}_{C_k})$, $k = 1, \ldots, K$ and the above computations are implementing the computation of the precision matrix of tree distribution $p(\boldsymbol{x})$ form its optimal marginals $q(\boldsymbol{x}_{C_k})$. It also explains why correlations for which $(i, j) \notin G(\boldsymbol{f})$ are omitted when computing the approximation.

3.3. Inference schemes and scheduling options

The choice of the inference scheme, determined by $G(\boldsymbol{f})$, and the scheduling of the updates in the fixed point iteration in (3.20)-(3.23) govern the accuracy and the speed of the inference algorithm. In the following we detail our choices and show how these correspond to well known approaches to inference in or model.

Based on the complexity of Collapse$[\cdot; \boldsymbol{f}]$ consider three main classes for $G(\boldsymbol{f})$: (i) *full*, where $G(\boldsymbol{f})$ is a fully connected, this corresponds to the classical approximate inference approach of propagating multivariate Gaussian (ii) *chordal*: where, $G(\boldsymbol{f})$ is chordal graph, corresponding to the propagation of messages having precision matrices with restricted sparsity structured, and (iii) *diag*, where $G(\boldsymbol{f})$ is a disconnected graph and thus only marginal means and variances are propagated.

In terms of scheduling we differentiate the following choices: (i) *static*, where the forward backward updates (3.20)-(3.21) are iterated until convergence and then an (3.23)-(3.23) update is performed, (ii) *sequential*, where the (3.23)-(3.23) updates are iterated until convergence followed by the corresponding (3.20)-(3.21) updates, and (iii) *dynamic*, where in order to minimise the number of update steps we use greedy scheduling strategy, to be detailed later.

$G(\boldsymbol{f})$ *is fully connected* (*full*). In this case both $\boldsymbol{Q}_{\boldsymbol{\alpha}_t}$ and $\boldsymbol{Q}_{\boldsymbol{\beta}_t}$ are full matrices implying that the diagonal blocks are $\boldsymbol{Q}_t^g$ are full and thus we are no longer dealing with sparse matrices. However, the *sequential* version of this algorithm corresponds to the classical filtering-smoothing approach with local approximations of the non-Gaussian terms and is a state of the art method in many statistical and machine learning applications. The *static* version of this algorithm can be shown to the expectation propagation in a block Gaussian model: the Gaussian forward-backward message passing corresponds to computing the marginal means and variances needed for the (3.23)-(3.23) updates. However, by considering the model as a latent block Gaussian model one can still preserve sparsity and apply the expectation propagation approach in [2] and thus replacing the forward-backward message passing with a partial matrix inversion on a $n \times T$ sized sparse matrix. Due to the former connection, the approximation provided by the latter two methods are identical.

$G(\boldsymbol{f})$ *is partially connected* (*chordal, tsp*). The *chordal* case is computationally less intensive than the *full* because it can preserve sparsity. This is both due to the sparsity of in $\boldsymbol{Q}_{\boldsymbol{\alpha}_t}$ and $\boldsymbol{Q}_{\boldsymbol{\beta}_{t+1}}$ and thus of $\boldsymbol{Q}_t^g$ and the computational cost of the of (3.25) and (3.26), which is much less than the $O(n^3)$ complexity of the *full* case. A special chordal case, when Collapse$(\cdot;\boldsymbol{f})$ is $O(n)$, is when we choose $G(\boldsymbol{f})$ as the maximum weight spanning tree of $|\boldsymbol{A}|$—we call this `tsp`. We detail the choice of chordal graphs in Section 3.4, however, typically one can assume that the computational complexity of the partial matrix inversion of $\boldsymbol{Q}_t^g$ and the collapse operations scales as $O(n^2)$, that is, in the worst case we expect $\sqrt{n}$ cliques of size $\sqrt{n}$. Operating on sparse matrices results in significant gains in terms of computational time. Although one expects that due to the "less informative" messages more update steps are needed, there is a very significant overall gain in speed as shown in Section 4. This clearly comes at the price of obtaining approximations that are less accurate than in case of *full*, however in many large scale applications this loss is not vey significant and one ends up with an overall advantageous compromise. The *static* scheduling option can be viewed as using as replacing the forward-backward to compute the marginal means and variances with an approximate (faster) forward-backward algorithm. In the *sequential* case there is a very significant computational gain since the linear algebraic operations are now carried out on sparse matrices.

$G(\boldsymbol{f})$ *is disconnected* (*diag*). This case corresponds to the inference scheme where the temporal messages are factorised. When $G(\boldsymbol{f})$ is disconnected, $\boldsymbol{Q}_{\boldsymbol{\alpha}_t}$ and $\boldsymbol{Q}_{\boldsymbol{\beta}_{t+1}}$ are diagonal, therefore, the temporal messages add no computational cost. Collapse$[\cdot;\boldsymbol{f}]$ simplifies identifying marginal means and variances, that is, it the same as Collapse$[\cdot;\boldsymbol{g}]$. As a result the partial matrix inversion of $\boldsymbol{Q}_t^g$ is extremely fast and Collapse$[\cdot;\boldsymbol{f}]$ comes at no computational cost. The computational cost scales with that of a sparse Cholesky factorisation and thus we expect it to scale as $O(n\log(n)^3)$. The *static* and *sequential* scheduling strategies have the same properties as in the *chordal* case.

The computation is dominated by the number of partial matrix inversions of $\boldsymbol{Q}_t^g$ and ideally, in all inference schemes, we would like to design a message-passing algorithm which achieves convergence with a minimal number of partial inversions steps. Clearly, this is not a straightforward problem to solve. For this reason, we propose a greedy *dynamic* scheduling where at every step we select the message that has the largest (last) update, and update both the receiver and the source of this message be it either $q_{t,t+1}^g$ or $q_{t+1,j}^l$. This is implemented by (i) keeping track of the updates in each $\boldsymbol{\alpha}_t, \boldsymbol{\beta}_t, \boldsymbol{\lambda}_{t+1,j}^g$ and $\boldsymbol{\lambda}_{t+1,j}^l$ and (ii) updating all the outgoing messages when updating an approximate marginal density. The computational saving due to the greedy *dynamic* scheduling are shown on the right panel of Figure 7. By constructing longer scheduling queues (ranking the updates) one can distribute the computation to several processing units and achieve a further reduction of computational time.

3.4. Chordal $G(\boldsymbol{f})$s and matrix re-orderings

The computational time is dominated by the partial inversion of $\boldsymbol{Q}_t^g$. The complexity of the computation is determined by the choice of $G(\boldsymbol{f})$. As detailed in Section 3.3 the structure of $\boldsymbol{Q}_t^g$ can range from full in case of *full* to a minimally sparse structure in case of *diag*. In this section we motivate our choices of chordal $G(\boldsymbol{f})$s. We start from the intuition that $G(\boldsymbol{f})$s that reflect the spatial connectivity (finite element grid structure) of the model would be good candidates for the precision structure of the messages. Note, that as shown in Section 2 and illustrated on Figure 1, this grid structure is identical to the sparsity structure $S(\boldsymbol{A})$ of the state transition matrix $\boldsymbol{A}$. Unfortunately, $S(\boldsymbol{A})$ is not a chordal graph. In order to define a chordal $G(\boldsymbol{f})$, we propose to complete $S(\boldsymbol{A})$ to a chordal graph by adding the least possible number of additional edges. These additional edges might not have direct intuitive meaning in terms of the model structure, however, their use makes inference work and minimises the computational effort.

It is well know that the sparse Cholesky factorisation creates sparsity structures that correspond to chordal graphs and that the structure of these graphs as well as the number of additional, so called, "fill in" edges depends on the row-column reordering methods applied prior to the factorisation [4]. These sparsity structures of the factors are computed before any numerical evaluations take place and are referred to as symbolic Cholesky factorisation. Depending on the sparsity structure of the matrix to be factored, various row-column reorderings have been proposed to minimise the number of "fill-inn"-s. We will make use of these properties of the factorisation to complete $S(\boldsymbol{A})$ to a chordal graph. Suppose σ is a reordering to be chosen later and that $S(\boldsymbol{L})$ is the symbolic Cholesky factor of $S(\boldsymbol{A}_{\sigma,\sigma})$, the we will use the chordal graph $G(\boldsymbol{f}) = [S(\boldsymbol{L}) + S(\boldsymbol{L}^T)]_{\sigma^{-1},\sigma^{-1}}$ to define our expectation constraints and the precision structure of the forward and backward messages. There are several well-known "fill-in" reducing reordering permutations, in this paper we will use (i) the approximate minimum degree (`amd`) permutation, (ii) the symmetric reverse Cuthill-McKee (`rcm`) permutation, and (iii) the nested dissection (`nd`) permutation.

When performing the Cholesky factorisation and partial matrix inversion of $\boldsymbol{Q}_t^g$, we consider the same permutation algorithms. We do an empirical estimation of the computational complexity of the Collapse$[q_{t,t+1}^g; \boldsymbol{g}]$ and Collapse$[q_{t,t+1}^g; \boldsymbol{f}]$ steps and we choose the best performing pair of permutations. In the models we considered the best performing pairs were the `amd` and `nd` permutations to obtain $G(\boldsymbol{f})$—from $S(\boldsymbol{A})$—and `amd` for the factorisation and partial inversion of $\boldsymbol{Q}_t^g$. The latter operation typically dominated the computational time and `amd` outperformed the other methods. `amd` and `nd` led to structures in $G(\boldsymbol{f})$ that resulted in similar computational times (clique size distribution, see Section 3.2) in the Collapse$[q_{t,t+1}^g; \boldsymbol{f}]$ step. If none of the above mentioned or any other chordal completion strategies are satisfactory, one can revert to a different $G(\boldsymbol{f})$ or perform the Collapse$[q_{t,t+1}^g; \boldsymbol{f}]$ as most appropriate in his/her model's context, see Equation (3.19).

The panels of Figure 5 the sparsity structures of $G(\boldsymbol{f})$, $\boldsymbol{Q}_t^g$ and the corre-

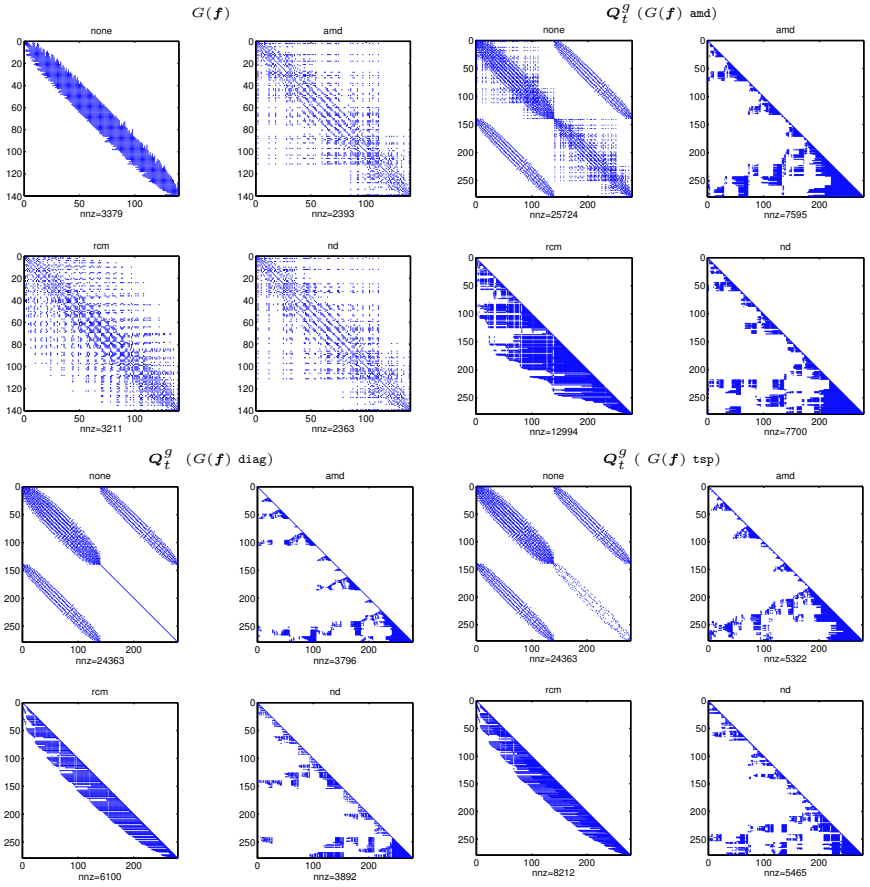

FIG 5. *In illustration of the sparsity structures of the matrices* $G(\boldsymbol{f})$ *and* $\boldsymbol{Q}_t^g$ *on a grid model similar to the one on Figure 1. The panels on the* $G(\boldsymbol{f})$ *block show the chordal completions of* $S(\boldsymbol{A})$ *(shown on the off-diagonal of* $\boldsymbol{Q}_f^g$*) obtained by symbolic Cholesky factorisations with various reorderings. The rest of the panels show the structure of* $\boldsymbol{Q}_t^g$ *for the corresponding choice of* $G(\boldsymbol{f})$ *as well as the structure of Cholesky factors for various reordering applied to* $\boldsymbol{Q}_t^g$*.*

sponding Cholesky factor for various choice of reordering permutations for a grid structure similar to that in Figure 1. The $G(\boldsymbol{f})$ group of panels show that no reordering generates a significantly less sparse than the ones using reordering. The `amd` and `nd` reorderings seem to generate structures with a similar number on non-zero elements. Note that $S(\boldsymbol{A})$ is shown on the off-diagonal block of the $\boldsymbol{Q}_t^g$ panels. The rest of the panels show the structure of $\boldsymbol{Q}_t^g$ and its corresponding Cholesky factors for the `amd`, `rcm`, and `nd` reorderings. Again the `amd` and `nd` reordering are shown to lead to a similar number of non-zero elements. Note that `diag` leads to structures that are significantly sparser and the average

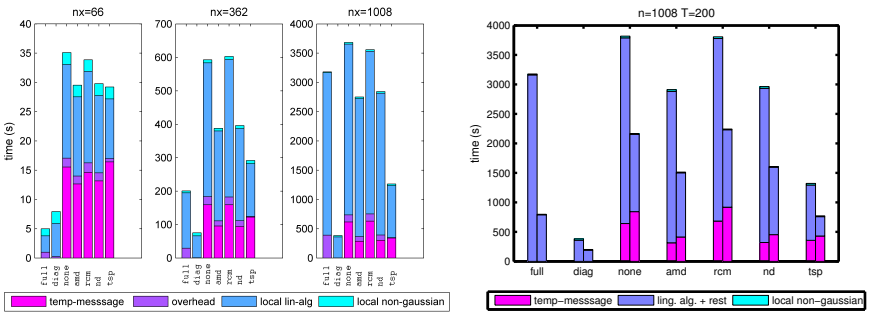

FIG 6. *Running times for various state space sizes and scheduling options. (left) Running times for the inference schemes* `full`, `diag`, *and chordal schemes* `none`, `amd`, `rcm`, `nd` *and* `tsp` *(left panel). Comparison in running times of the sequential (left bar) and greedy (right bar) scheduling strategies. Local operations refer for the local linear algebra whilst the temporal messages refer to the max-determinant optimisation (right panel).*

number of non-zeros per column in the Cholesky factor is much smaller than in any other case. This has a significant impact on the performance of the partial matrix inversion by the Takahashi equations which scale as $\sum_j (\sum_{i:L_{i,j}\neq 0} 1)^2$ [e.g. 24, 2].

4. Experiments

In this section we assess the running times and accuracy of the inference methods we introduced and show the potential use of this approach through the `diag` method in the WikiLeaks Afghan War Diary data studied in [28]. The algorithms were coded in Matlab and we used the partial matrix inversion algorithm of [7] which is implemented in the C programming language.

4.1. Accuracy

To assess the accuracy of the different methods we constructed a 1-D grid toy model with $n = T = 64$, where at each time point the hidden state at a grid point becomes an average of itself and its 5 neighbours on either side. Hence the matrix $\boldsymbol{A}$ is banded with bandwidth 5. Each $\boldsymbol{e}_k$ was sampled with an exponentiated squared covariance with variance 1 and length scale of 5 units (neighbours) - this describes a correlation which decreases from 1 to 0.1 in 3.4 spatial units. The algorithms were assumed to have converged when the maximum absolute change in canonical parameters became less than 10^{-4}.

We generated 150 realisations of this model and assessed the quality using

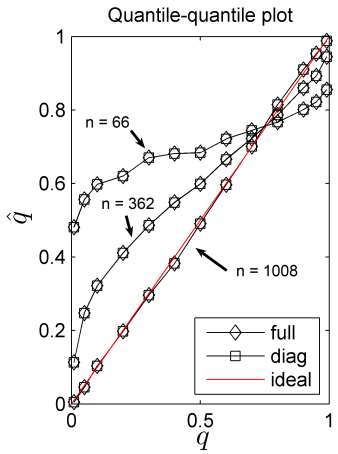

FIG 7. *Quantile-quantile plots for* `full` *and* `diag` *with* $n = 66, 362, 1008$ *in increasing order of accuracy (right panel).*

two criteria, (i) the mean total square error per time frame (MTSE) defined as

$$MTSE = \frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{n}(x_t^{j^*} - \hat{x}_t^j)^2$$

where $\boldsymbol{x}_t^*$ is the simulated realisation and $\hat{\boldsymbol{x}}_t$ is the marginal mean at time t, and (ii) the quantile-quantile bias, defined as

$$bias_q = q - \frac{1}{Tn}\sum_{t=1}^{T}\sum_{j=1}^{n}\mathbf{1}[x_t^{j^*} \leq x_t^{j^q}]$$

where $\mathbf{1}[\cdot]$ is the indicator function and $x_t^{j^q}$ is the inferred threshold of quantile q of the approximate marginal, see [25] for details. Friedman tests for repeated measures [6] found that there were significant differences in both the quantile bias and the MTSEs in the proposed methods ($p < 0.01$). Post-hoc tests showed that `diag` and the `tsp` are, as expected, less accurate than the `chordal` and `full` and that there is no significant difference ($p > 0.2$) between the latter two in both MTSE and bias. The `tsp` only slightly outperformed the `diag`, see Table 1.

Whilst it is hard to generalise conclusions regarding the performance, we do note a slight decrease in performance of `diag` and `tsp` with increasing correlations induced by $\boldsymbol{Q}$ and $\boldsymbol{A}$; however, in all cases the difference proved to be small relative to the overall performance of the algorithms. It should be noted that in this example a damping $\epsilon = 0.75$ was applied on the temporal messages of `diag` to ensure convergence; because of this it was slower than the `full` and

	`full`	`chordal`	`tsp`	`diag`
$\Delta MTSE$	0	0.000	0.017	0.018
Δq^{bias}	0	3×10^{-6}	3×10^{-4}	3×10^{-4}

TABLE 1
Algorithmic performance with `full` *as reference*

`chordal` schemes. The good performance of the `chordal` is also due to the fact that the banded structure is (already) chordal.

4.2. Running times and scalability

To compare running times, we choose a typical scenario where the latent field is governed by

$$\begin{aligned} \mathrm{d}z(\boldsymbol{s},t) &= \mathcal{A}z(\boldsymbol{s},t)\mathrm{d}t + \mathrm{d}W(\boldsymbol{s},t), \\ z(\boldsymbol{s},0) &= z_0(\boldsymbol{s}), \end{aligned} \tag{4.1}$$

with $W(\boldsymbol{s},t)$ being a space-time Wiener process having a covariance operator $\Sigma u(\boldsymbol{s}) = \int k(\boldsymbol{s},\boldsymbol{r})u(\boldsymbol{r})\mathrm{d}\boldsymbol{r}$. The operator $\mathcal{A} = D\Delta(\cdot)$ where $\Delta(\cdot)$ is the Laplacian and we used a circular domain for $\mathcal{O}$ with radius r. Temporal discretisation with $\Delta_t = 0.01$ followed by the Galerkin method in conjunction with a row-sum lumping method [1, 13] was applied to obtain a sparse matrix $\boldsymbol{A}$ with $n = 2267$ for simulating data with $T = 200$. We set $r = 10, D = 0.2$ and assumed that the discretised precision matrix $\boldsymbol{Q}$ is diagonal with elements $1/15$.[2] We found that under this configuration, at stationarity, around 1000 points per time frame were generated; a typical count for large data sets.

The algorithms were tested on Delaunay triangulations of the domain constructed using routines by [19] with varying mesh density, $n \in \{362, 562, 1008\}$. Computational times were recorded using Matlab's profiler. In both the static and sequential scheduling schemes, $\boldsymbol{\lambda}^{gs}_{t+1,\cdot}$ and $\boldsymbol{\lambda}^{ls}_{t+1,\cdot}$ messages were run till convergence. To ensure a fair comparison, all test results given here are with computations restricted to a single processor core.[3]

The computational times for the sequential scheduling are plotted in the left panel of Figure 7. We segmented the computational times to correspond to the three main collapse operations: (1) *temporal messages* stands for the max determinant optimisation problem (overhead in `full`) (2) *overhead* accounts for initialisations, updating messages and monitoring convergence (3) *lin-alg* logs the time for linear algebraic operations (Cholesky factorisation, partial matrix inversion), and (4) *non-Gaussian* stands for the univariate moment matching. For clarity, we omit results for the static case which was up to an order of magnitude slower than the second worst-performing method. This slow performance

[2] This follows when modelling the covariance function $k(\boldsymbol{s},\boldsymbol{r}) = \sum_i \phi_i(\boldsymbol{s})\phi_i(\boldsymbol{r})\tilde{\lambda}_i = \boldsymbol{\phi}(\boldsymbol{s})^T\boldsymbol{\Lambda}\boldsymbol{\phi}(\boldsymbol{r})$.

[3] All algorithms were tested on an Intel® Core™i7-2600S @ 2.80GHz personal computer with 8GB of RAM.

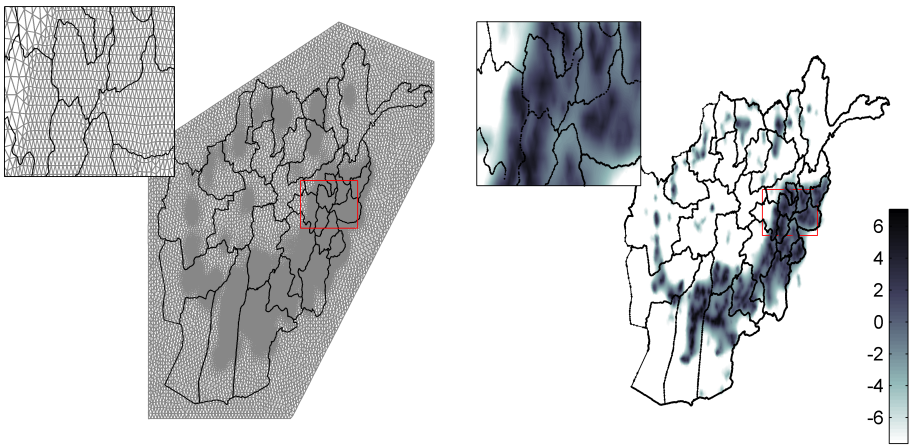

FIG 8. *The mesh and one time-slice log intensity map corresponding to the AWD on the first week of October 2009.*

is attributed to the number of iterations required for convergence, which is more than the number of forward-backward cycles required by the `full`.

The left panel in Figure 7 shows that, for small n, the `full` inference scheme is faster than the other schemes due to the fact that it is implemented more efficiently in terms of matrix operations. However, the situation changes for large n and for $n = 1008$, where we see that the `full` is slower than the best `chordal` methods and much slower than the `tsp` and `diag`. Note that the increase in computational time is well below cubic and at most quadratic for all methods other than the `full`. We could not run the `full` for larger n due to memory restrictions. As shown in the centre panel of Figure 7, in this case the increased speed does not come at a loss in distributional accuracy and despite the data being simulated at $n = 2267$ quantification of uncertainty at $n = 1008$ is very good for all methods. This was unexpected given the correlations induced by $\boldsymbol{A}$; at this stage it is not yet clear to which extent the performance of the `diag` varies with the parameters of the system.

A small note is due on the speed-ups made possible when using greedy scheduling, as opposed to sequential scheduling (see right panel in Figure 7). Although the scheduling itself does not affect the scaleability of the algorithm, it can be seen that, as expected, the greedy scheduling can drastically reduce the computational time. For instance, after the initial forward-backward, the `full` needed only a few factor updates to achieve convergence within tolerance.

4.3. The Afghan War Diary

Zammit-Mangion et al. (2012) introduced point-process modelling methods for conflict and employed an algorithm similar to the `full` described here in an

iterative state-parameter update scheme on the WikiLeaks Afghan War Diary (AWD). However modelling of micro-scale effects such as relocation or escalation diffusions in conflict [22] were not possible at the resolution considered; whilst an average basis there had a scope on the order of 100km, conflict diffusions are observed at resolutions of ≈10km. The scope here is to show that we can perform inference on the required spatial and temporal scales.

We assembled a mesh on Afghanistan using population density as a proxy for mesh density. The resulting construction, shown in Figure 8 has the largest triangles with sides of 22km and the smallest ones with sides of 7km. The total number of vertices amounts to $n = 9398$ in a system with $T = 313$ time points (weeks). We constructed $\boldsymbol{A}$ from the diffusion equation above with $D = 1 \times 10^{-4}$ with latitude/longitude used as spatial units. $\sigma_w^2 = 0.2$ was taken as rough value from the full joint analysis using a low resolution model, see [28] for details.

We carried out inference in the AWD with the `diag` algorithm. A characteristic plot showing one week of the conflict progression (first week of October 2009) is given in Figure 8. At this point in the conflict activity in the south in Helmand and Kandahar was reaching its peak and conflict at the Pakistani border was intensifying considerably. The insets show how detailed inferences can be made - note that here we have employed fixed hyper-parameters; spatially-varying smoothness as in [13] may be introduced in the model with relative ease. Inference completed in just over 5 hours on a standard PC and consumed only about 4GB of memory. This performance implies that, with appropriate exploitation of clustered/distributed computational resources, full state-parameter inference of very large systems can be carried out in considerably shorter timescales and with potentially less resources than previously envisioned.

5. Conclusions

In this paper we have presented a family of EP inference methods for spatio-temporal log-Gaussian Cox process models; the algorithms are based on approximate inference methods using expectation constraints. We show how the sparsity in the underlying dynamic model can be exploited in order to overcome the limitations in the standard forward-backward and block inference schemes which can become expensive w.r.t. both storage and computation for large n and T. Our approach is applicable in any similar sparse latent linear dynamical model. For the models we studied, `diag` is faster but less accurate than `chordal`. The inference schemes using messages with chordal precision structures can serve as a compromise in complexity between schemes using diagonal and full precision matrix structures.

In the future we intend to explore the set of structures that lie between the fast and less accurate spanning tree and the somewhat larger chordal structures employed in this work. We are currently working on including parameter inference and correction methods [2] in the methodological framework; this will be addressed in a follow-up paper.

References

[1] Bueche, D., Sukumar, N. and Moran, B. (2000). Dispersive properties of the natural element method. *Computational Mechanics* **25** 207–219.

[2] Cseke, B. and Heskes, T. (2011). Approximate marginals in latent Gaussian models. *Journal of Machine Learning Research* **12** 417–454.

[3] Dahl, J., Vandenberghe, L. and Roychowdhury, V. (2008). Covariance selection for nonchordal graphs via chordal embedding. *Optimization Methods Software* **23** 501–520.

[4] Davis, T. A. (2006). *Direct Methods for Sparse Linear Systems (Fundamentals of Algorithms 2)*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

[5] Diggle, P., Rowlingson, B. and Su, T. (2005). Point process methodology for on-line spatio-temporal disease surveillance. *Environmetrics* **16** 423-434.

[6] Field, A. (2009). *Discovering statistics using SPSS*. Sage Publications Limited.

[7] Gerven, M., Bahramisharif, A., Farquhar, J. and Heskes, T. (2012). Donders Machine Learning Toolbox.

[8] Hartikainen, J., Riihimäki, J. and Särkkä, S. (2011). Sparse spatio-temporal Gaussian processes with general likelihoods. In *Proceedings of the 21th International Conference on Artificial Neural Networks - Volume Part I. ICANN'11* 193–200. Springer-Verlag.

[9] Heskes, T., Opper, M., Wiegerinck, W., Winther, O. and Zoeter, O. (2005). Approximate inference techniques with expectation constraints. *Journal of Statistical Mechanics: Theory and Experiment*.

[10] Hooten, M. B. and Wikle, C. K. (2008). A hierarchical Bayesian non-linear spatio-temporal model for the spread of invasive species with application to the Eurasian Collared-Dove. *Environmental and Ecological Statistics* **15** 59-70.

[11] Kuss, M. and Rasmussen, C. E. (2005). Assessing Approximate Inference for Binary Gaussian Process Classification. *Journal of Machine Learning Research* **6** 1679–1704.

[12] Lauritzen, S. L. (1996). *Graphical Models. Oxford Statistical Science Series*. Oxford University Press, New York, USA.

[13] Lindgren, F., Rue, H. and Lindström, J. (2011). An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. *Journal of the Royal Statistical Society B* **73** 423-498.

[14] Minka, T. P. (2001). A family of algorithms for approximate Bayesian inference PhD thesis, MIT.

[15] Murphy, K. P. and Weiss, Y. (2001). The Factored Frontier Algorithm for Approximate Inference in DBNs. In *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence* 378-385.

[16] Opper, M. and Archambeau, C. (2009). The variational Gaussian approximation revisited. *Neural Computation* **21** 786–792.

[17] Opper, M., Paquet, U. and Winther, O. (2009). Improving on Expectation Propagation. In *Advances in Neural Information Processing Systems 21* (D. Koller, D. Schuurmans, Y. Bengio and L. Bottou, eds.) 1241–1248. MIT, Cambridge, MA, US.

[18] Opper, M. and Winther, O. (2000). Gaussian Processes for Classification: Mean-Field Algorithms. *Neural Computation* **12** 2655-2684.

[19] Persson, P. O. and Strang, G. (2004). A simple mesh generator in MATLAB. *SIAM review* 329–345.

[20] Rodrigues, A. and Diggle, P. J. (2012). Bayesian Estimation and Prediction for Inhomogeneous Spatiotemporal Log-Gaussian Cox Processes Using Low-Rank Models, With Application to Criminal Surveillance. *Journal of the American Statistical Association* **107** 93-101.

[21] Rue, H., Martino, S. and Chopin, N. (2009). Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal Of The Royal Statistical Society B* **71** 319-392.

[22] Schutte, S. and Weidmann, N. B. (2011). Diffusion patterns of violence in civil wars. *Political Geography* **30** 143–152.

[23] Simpson, D., Illian, J., Lindgren, F., Sørbye, S. and Rue, H. (2011). Going off grid: Computationally efficient inference for log-Gaussian Cox processes. *arXiv preprint arXiv:1111.0641*.

[24] Takahashi, K., Fagan, J. and Chin, M. S. (1973). Formation of a sparse impedance matrix and its application to short circuit study. In *Proceedings of the 8th PICA Conference*.

[25] Taylor, B. M. and Diggle, P. J. (2012). INLA or MCMC? A Tutorial and Comparative Evaluation for Spatial Prediction in log-Gaussian Cox Processes. *arXiv preprint arXiv:1202.1738v2*.

[26] Wikle, C. K. (2002). A kernel-based spectral model for non-Gaussian spatio-temporal processes. *Statistical Modelling* **2** 299-314.

[27] Zammit-Mangion, A., Sanguinetti, G. and Kadirkamanathan, V. (2012). Variational estimation in spatiotemporal systems from continuous and point-process observations. *IEEE Transactions on Signal Processing* **60** 3449–3459.

[28] Zammit-Mangion, A., Dewar, G. M., V., K., A. and Sanguinetti, G. (2012). Point process modelling of the Afghan War Diary. *Proceeding of the National Academy of Sciences*.